UNIVERSITY OF QUEENSLAND
ST.LUCIA
QUEENSLAND
AUSTRALIA 4067

PRENTICE COMPUTER CENTRE

NEWSLETTER

N-231 17JUL78

CONTENTS

authorization: Director of the Centre

1 NEW SPECIAL INTEREST GROUP - MATHEMATICAL AND STATISTICAL PACKAGES

The following is a text of a letter forwarded to Heads of Departments University of Queensland and Chairmen of Schools Griffith University. External clients of the Centre making use of these packages are also invited to join the special interest group:

> Over recent years the Centre has acquired and supported a range of packages in the mathematical and statistical area for operation on the central computing equipment. There are further demands from user groups to add to the range of these packages. Similarly, with the acquisition by a number of departments of PDP11 minicomputers, there is an interest that the Centre acquire and provide support for such packages in the minicomputer area.
>
> These packages are used substantially in both teaching and research in the University and it is felt that the stage has been reached where there would be value, both to the user group and the Centre, in the formation of a Special Interest Group. The activities of such a group would eventually be determined by the group itself but would probably include the following:
>
> (a) Advice to the Computer Centre on
>
> (i) the need for acquisition of new packages;
>
> (ii) the rationalization and priorities in support of existing packages;
>
> (iii) type and level of support required including training, audio visual material etc.
>
> (b) Sponsorship of studies relating to validity and relative efficiency and appropriateness of routines in one package as compared to others.
>
> (c) General interchange of information between users, the Centre, and other Universities and research organizations.
>
> Dr. John Holt of the Department of Mathematics has kindly consented to help the Computer Centre with the establishment of this activity.
>
> I would be grateful if you would please bring this matter to the notice of interested staff in your Department. Those wishing to join the Special Interest Group on Mathematical and Statistical Packages might please advise the Centre's Secretary (Miss J. Dixon extension 2189) and they will be advised details of a first meeting of the Group.
>
> Director (2189)

2 FORTRAN

Much of this newsletter is concerned with Fortran. There is a new maintenance release of F10, a discussion of double precision on the KL10, advice of the need to move to the standard DEC device table and a recommendation to convert programs from F40 to F10.

Ideally, I would have preferred to produce a special issue of the newsletter on Fortran including extracts of continuing relevance from previous newsletters. Unfortunately, this was not possible in the time available but I feel it would be a useful document and we will produce it in the not too distant future.

Director (2189)

3 F40 - CHANGE IN SUPPORT CATEGORY

"FORTRAN 40 is a product which has been a part of the DECsystem-10 software family almost from the beginning. It is now extremely stable and over ten years old. FORTRAN 40 has been succeeded by six versions of FORTRAN 10 and has remained relatively inactive in recent years. It has had almost no performance problems in the past few years.

It is for these reasons that Digital feels it appropriate to change the support category of FORTRAN 40 from B to C. This means this time-proven product will remain available for purchase but will no longer receive support from the Software Support Group.

This revision in support category will be effective three months from now, on July 1, 1978. We are confident that this adjustment will not significantly impact your operations. "

Software Dispatch 1 April 1978

This announcement from DEC confirms a long obvious fact that the version of 1/75 which we currently use is the last release of F40. It will remain on the system into the foreseeable future but we will not be able, as DEC are not able, to give it full support; thus according to our support classification, it will revert to class 3, i.e. available 'as is'.

It is worth commenting that in every case where problems were encountered with Fortran compilers, F10 was found to give at least the same or better results.

Following this note is a reprint of an article submitted by a DECUS user on the conversion of F40 programs to run with F10. Some parts are perhaps not quite applicable, e.g. the reference to the routine ASSDEV, but we would encourage you to convert programs to use F10.

C.C. de Voil (3023)

KENTRON HAWAII, LTD.
DOT-ADP Support Services Project

DEPARTMENTAL CORRESPONDENCE

Subject: How to Convert from F40 to F10

Date: February 16, 1977

Memo No: K-4A-M104E/77

To: Scientific Section

From: C. Faylor/KHL

Copy to: S. Moffatt/KHL
C. O'Connor/KHL
R. Peabody/862
M. Todd/KHL
R. Wassmuth/KHL

F40 is scheduled to be taken from the DEC-10 sometime after March. This memo describes most of the differences between F40 and FORTRAN-10. It is intended to be used as an aid in the conversion from F40 to FORTRAN-10.

1. ASSDEV, IFILE and OFILE can all be replaced with the OPEN statement in FORTRAN-10. For example, if a program contains the statement:

```
CALL ASSDEV(1,'TTY')
```

it can be replaced with:

```
OPEN(UNIT=1,DEVICE='TTY')
```

Or, if a program contains a sequence of statements:

```
CALL ASSDEV(1,'DSK')
CALL IFILE(1,'TEST')
```

they can be replaced with:

```
OPEN(UNIT=1,DEVICE='DSK',FILE='TEST',
ACCESS='SEQIN')
```

If the IFILE in the last example was an OFILE, then ACCESS='SEQOUT'

2. Binary files created by an F40 program do not work with F10. A program is available which converts binary files to F10 format from F40. (See S. Shapiro's memo no. K-4A-EM354E/77)

3. Logical constants cannot be abbreviated in FORTRAN-10.

```
Change    LOGIC  = .T.
          LOGIC1 = .F.
       to LOGIC  = .TRUE.
          LOGIC1 = .FALSE.
```

4. Subroutines with multiple returns act differently in FORTRAN-10. For instance, if a program contains the statement:

```
      CALL SUB(A,B,$100,$200,$300)
        .
        .
        .
100   CONTINUE
        .
        .
        .
200     N
        .
        .
        .
300   CONTINUE
        .
        .
        .
      END
      SUBROUTINE SUB(A,B,*,*,*)
        .
        .
        .
      RETURN 3
      END
```

F40 will return from the subroutine at statement 100, FORTRAN-10 will return at statement 300. The reason for this is that in F40 a RETURN i statement transfers control to the i^{th} argument in the subroutine call. FORTRAN-10 returns to i^{th} asterisk, (which should be the i^{th} statement label in a subroutine call).

```
Change    RETURN 3
       to RETURN 1
```

5. Any octal constants in a program should be preceded by a Double Quote in FORTRAN-10 rather than the letter O of F40.

```
Change    A=O12345677
       to A="12345677
```

6. Exponentiation is grouped from right to left in FORTRAN-10 instead of left to right in F40.

```
Change    A**B**C
       to A**(B**C)
```

7. Macro programs written for F40 will not work with F10. See page 0-18 of the FORTRAN-10 PROGRAMMER'S REFERENCE MANUAL, sixth edition for more information

8. The filename extension expected for FORTRAN-10 source files is .FOR instead of .F4.

```
To change:  RENAME FILE.FOR=FILE.F4
```

8. Any specification statements (like INTEGER, COMMON, DIMENSION) must appear at the beginning of the source program in FORTRAN-10.

```
Change    A=B
          INTEGER X
          X=A*B
          REAL I
          I=X

       to INTEGER X
          REAL I
          A=B
          X=A*B
          I=X
```

10. Any DATA statements must appear after all specification statements:

```
Change    DATA I,J/13,17/
          INTEGER A,B
          A=I

       to INTEGER A,B
          DATA I,J/13,/17/
          A=I
```

11. There are some differences in the way FORTRAN-10 and F40 handle I/O.

a) Consider the program:

```
      N=0
      WRITE(5,100)N
  100 FORMAT(1X,A5)
      END
```

When this program is executed under F40, a "NULL" will be output to the terminal (i.e. nothing is typed). FORTRAN-10 will output a "blank" to the terminal.

b) The ERR= option for READ/WRITE statements detects different types of errors in F10.

Consider the program segment:

```
        .
        .
        .
    READ(5,100,ERR=200)A
100 FORMAT(F5.1)
        .
        .
        .
200 CONTINUE
```

and assume the following was typed on the terminal:

```
12A.7
```

when the F40 program reads the non-numeric character "A", it transfers to statement 200. When the FORTRAN-10 program encounters the non-numeric character, it terminates execution. In FORTRAN-10, there is no simple way to check for an illegal character on input since the "ERR=" only detects hardware problems like parity errors rather than software problems like illegal characters.

12. The ENCODE/DECODE statements in F40 are not quite equivalent to their FORTRAN-10 counterparts.

Consider the program:

```
      A='HELLO'
      I=12
      ENCODE(5,100,I),I,A
100   FORMAT(I2,A3)
      TYPE 200,I
200   FORMAT (1X,A5)
      END
```

F40 does what might be expected when this program is run. It outputs 12HEL to the terminal.

Before the program is run in FORTRAN-10 however, the comma before the second I in the ENCODE

statement must be deleted. This comma was optional in F40. It is not in FORTRAN-10.

Output from the program when run under FORTRAN-10 will probably be something like:

"blank" "blank" HEL

With FORTRAN-10, the results are unpredictable when a variable is specified within the parenthesis of a DECODE/ENCODE statement and in the variable list.

```
Change    ENCODE(5,100,I),I,A
     to K=I
        ENCODE(5,100,I) K,A
```

13. On a final note, FORTRAN-10 handles all double precision computations with hardware instructions instead of software subroutines, as in F40. This means that double precision arithmetic works well in FORTRAN-10 where it worked poorly or not at all under F40.

Christopher Faylor

C. Faylor/KHL

CF/js

4 FORTRAN COMPILER RELEASE

On Wednesday 5 July, 1978, versions of the F10 compiler were changed as follows --

Version 5(515)	transferred from	STD:	to OLD:
Version 5A(621)	transferred from	NEW:	to STD:

The new release is a maintenance release only, incorporating in excess of 60 bug fixes which have been applied to the previous version. It has been in use on NEW: since early this year and is known to correct several reported errors.

It is not thought that any problems will be encountered in its use, though naturally it is not possible to be absolutely certain that some of the changes introduced will not have side-effects which will affect some usages. The file DOC:FTN5A.DOC is DEC's supporting release document for this version of the compiler and those interested in detailed changes should consult it.

In this document, the listed known bug/deficiency is --

--- Deeply nested expressions might get "stack overflow" in FORTB.

Additionally, in the file DOC:FTN5A.ERR are the problems which have been reported since the release of this version, up to the Software Dispatch of 1 July, 1978.

Chris de Voil (3023)

5 FORTRAN DOUBLE PRECISION ON THE KL10 - AN IMPORTANT NOTICE

As users may be aware, the new KL10 system has additional hardware instructions to improve performance in certain areas. One of these areas concerns the execution of double-word instructions, as for example are used for Fortran Double Precision operations.

Unfortunately, these new instructions assume differing formats for double-word variables and this has led to some incompatibilities between the default behaviour of the F10 compiler and the Fortran library -- FORLIB on STD:. To overcome this incompatibility which only affects programs using Double Precision variables, the following procedures should be adopted.

If you use only the standard software
Make sure that all programs which have been compiled on the KL10 are re-compiled with the switch KA10. This only affects programs which were compiled on the KL10 and may be done with a Load-Compile command such as -

.LOAD/COMPILE A.FOR(KA10) . . .

If this is not done, double precision operations will be performed with less than single precision accuracy and erroneous results will be obtained. Programs or subroutines which were compiled on the KA10 are quite satisfactory and will continue to work properly with the standard software.

If you customarily use NEW: software
Do nothing for any new work; you will be using a compatible F10 compiler and FORLIB.
However, if you have routines which have not been re-compiled since you were transferred to the KL10, recompile them.

In summary it is most important that all routines which go to make up a program are compiled to use the same code type, that is either KA10 code or KL10 code. If this is not done for routines using double precision arithmetic then incorrect results will be obtained, in other cases it is not so critical, but it is our recommendation that you recompile if there is any doubt.
When using STD: software always use the (KA10) switch for Fortran code.
When using NEW: make sure all routines have been compiled on the KL10.
You can use NEW: software by using the switch /NEW at Login, or by using the program SETSRC.

This problem arose because in transferring software from the KA10 to the KL10 we wished to minimize the perturbations users would experience by as far as possible setting up the KL10 with exactly the same software as was being used on system areas on the KA10. Unfortunately however, the F10 compiler checks which type of machine it is running on and by default creates the most appropriate code for it. Thus the same compiler on the KA10 will generate code using the KA10 double precision procedures, but on the KL10 its default will be to generate code utilizing the KL10 double word instructions. It is possible to over-ride this default with the switches KA10 and KI10 and that is what may have to be done in this case. (The switch is KI10 because these instructions were first implemented on the KI10 processor which is intermediate between the older KA10 and the KL10.)

Now for the good news.

Some tests have been made to evaluate the new double precision procedures and they really do have an advantage. It was only possible to compare the present STD: Forlib (which is version 5) with the NEW: version (V. 5A) so these comparisons are not entirely fair, but are adequate in that these two are the only versions of Forlib accessible to

users. A program was written to repeatedly invert in double precision mode a matrix which was initialized to be a Hilbert matrix of specified order. Using the KA procedures, the program was in fact just under 500 words smaller than when the KL instructions were used (note though, that both versions were run on the KL10 for this test). Tabulated below against the size and number of inversions performed is the cost in cents and the average error of all terms in the resultant matrix.

Matrix Size	No. of Inversions	STD: system KA10 Insts. Cost	Ave. Error	NEW: system KL10 Insts. Cost	Ave. Error
7	400	236	.13D-06	119 (109 optimized)	.14D-10
10	200	344	.33D-02	168	.30D-07

Briefly, while this is a special case, it would appear that better accuracy is obtained and the cost is about half as compared to the KA procedures. It must be emphasized that this is for a program using a great deal of double precision arithmetic and the same advantage may not be achieved in other programs.

However there are some potential problems associated with the use of the NEW: version.

In the interests of improved maintainability, the NEW: version has the DEC standard device table which is used to allocate devices to Fortran Logical Units in the absence of OPEN statements in the program or Monitor ASSIGN commands. The differences (which are underlined) are-

Fortran Logical Unit	Present U Q Device	Standard DEC Device
5	TTY	TTY
6	TTY	PTR
8	DTA0	DIS
10	DSK	DTA2
11	DSK	DTA3
12	DSK	DTA4
13	DSK	DTA5
14	DSK	DTA6
20 to 24	DSK	DSK

The changes to the device table were made for operation in a different environment and with a different Fortran Operating system. Facilities now exist in Fortran through use of the OPEN statement to exercise much better control over device assignment than was formerly possible and experiments have shown that in practically every case, existing programs can continue to be run properly with appropriate Assign commands. Naturally though, there would be an advantage in making the necessary slight changes to the program at source level so that the Assign command was not necessary.

A change to the Device Table was foreshadowed several years ago and over a year ago an alternative form was installed on NEW. experimentally but finally not proceeded with. For the reasons of the cost advantage indicated above and the matter of maintainability, it would seem desirable to implement this version as standard as soon as convenient. However, we realize that some users could be inconvenienced by too sudden a change and so it is our intention to make this version standard over the Christmas break. The version of Forlib and Forots installed on NEW: on 25 July, 1978 includes all reported error patches from the release of this version up to the Software Dispatch of 1 July, 1978. It also includes the Fortran Sort Module as described in the Sort/Merge User's Guide.
If users wish to experiment or run their programs under the NEW: system, they should include NEW in their Search Lists. This may be done with the switch NEW at Login, or by using the program Setsrc.
The file DOC:FRS5A.DOC is the supporting DEC release report for this new system. It has been edited to include the changes made by this installation.

We would be pleased to have comment from any users on this subject.

Chris de Voil (3023)

6 IDENTIFYING YOUR TERMINAL

Terminal or tty numbers on the DEC1090 are not constant for any terminal. This is unlike the behaviour of the DEC1055 where a particular terminal port to the system was always assigned the same number e.g. TTY30.

Each terminal port on the DEC1090 is described by a pair of numbers: the node number and line number within that node. The actual terminal number assigned to a specific port depends on the order in which other network nodes have started. The terminal number should not be expected to remain constant from day to day. The node and line number pair will remain constant though for a specific terminal port.

The following skeleton MACRO code illustrates one possible way of calculating node and line number for the terminal controlling the job:

```
        search   uuosym

;returns with t1 containing node number in left-half
;line number in right half.  If controlling terminal
;is not a network terminal, the left half is set to zero
;and the right half contains the terminal number.  Returns
;with t1 negative if no controlling terminal.
;call via
;         PUSHJ   P,TTY2NL

          ENTRY   TTY2NL

TTY2NL:   getlin    t1,        ;get name of controlling terminal
          gtntn.    t1,        ;convert to node and line number
           skipa                ;error, try another way
          popj       p,        ;successful, return with answer
          seto      t1,        ;-1 for this job
          trmno.    t1,        ;get controlling terminal io index
           setz     t1,        ;no controlling terminal
          subi      t1,.uxtrm  ;convert to line number
          popj       p,        ;return with answer.
```

Arthur Hartwig (3021)

7 NEW VERSION OF LINK

Version 4 of LINK is now on NEW:. Apart from performance improvements, the main differences are that /SAVE and /SSAVE now write a .EXE file, and error messages are improved to correspond with those in the manual. The documentation file is DOC:LINK.DOC.

This version of LINK will be moved to STD: on 1 September, at which time the version currently on STD: will be moved to OLD:.

Will Gout (3023)

8 REMOVAL OF OFFLINE FILES

Since 1975, a number of files that had not been accessed for some time have been deleted from the offline areas and put onto magnetic tape. Listings of these file names have been forwarded to departmental secretaries for distribution. For any further information please telephone the Operations Supervisor on extension 3212.

9 DIABLO TERMINAL SUPPLIES

If owners of Diablo terminals are interested in bulk purchase of Ribbons and Print Daisy Wheels, would they please contact Dal Anderson (extension 3166).

* * * * *